

Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan *Node.js*

Muhammad Iqbal C. R., Muchammad Husni dan Hudan Studiawan
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: husni@its-sby.edu

Abstrak—Dewasa ini, pengembangan dan pemanfaatan teknologi jaringan dan komunikasi mengalami perkembangan yang pesat. Perkembangan teknologi yang sangat pesat saat ini mendorong internet dan web untuk menjadi standar komunikasi yang terbaru. Dengan diperkenalkannya konsep Web 2.0 dan standar HTML5, web diharapkan dapat menjadi standar komunikasi yang universal. Di sisi lain, *Rich Internet Application* (RIA) menjadi tren perkembangan teknologi web saat ini. Dengan teknologi tersebut, aplikasi web didesain dengan pendekatan aplikasi desktop, mengedepankan kecepatan akses, kemudahan bagi pengguna, dan keinteraktifan. Sehingga, membawa teknologi komunikasi VoIP ke teknologi RIA akan semakin memudahkan pengguna. Dengan mengimplementasikan fitur-fitur baru dari HTML5, akan dihasilkan aplikasi web yang mengikuti konsep RIA tanpa memerlukan penggunaan aplikasi pengaya pada peramban. Untuk mengetahui kemampuan dari sistem ini dilakukan tiga pengujian. Pengujian pertama adalah untuk mengetahui tingkat QoS (*Quality of Service*) dari perangkat lunak dan yang kedua adalah survei untuk mengetahui nilai MOS (*Mean Opinion Score*) dari perangkat lunak. Survei menghasilkan keluaran bahwa perangkat lunak memiliki skor MOS 3,8. Pengujian ketiga adalah untuk mengetahui kompatibilitas dari aplikasi-aplikasi peramban yang populer saat ini, hasil dari pengujian ini hanya peramban Google Chrome Developer Release yang mendukung aplikasi web ini secara penuh. Dari hasil implementasi yang telah dilakukan, penggunaan teknologi HTML5 dapat terbukti diimplementasikan dalam membuat aplikasi VoIP berbasis web, walaupun terdapat masalah kompatibilitas di mana fitur WebRTC belum diimplementasikan pada peramban-peramban yang umum digunakan.

Kata Kunci— HTML5, *Node.js*, SIP, VoIP, Websocket.

I. PENDAHULUAN

TEKNOLOGI Voice over Internet Protocol (VoIP) kini berkembang sebagai salah satu metode komunikasi yang layak diperhatikan. Teknologi ini sering dipertimbangkan sebagai sebuah alternatif metode komunikasi karena memiliki keunggulan-keunggulan dibandingkan *Public Switched Telephone Network* (PSTN) yang ada saat ini. Keunggulan-keunggulan tersebut antara lain biaya penggunaannya yang lebih rendah dan fungsionalitas yang lebih banyak [1].

Konsep *Rich Internet Application* (RIA) adalah sebuah konsep dalam pembuatan aplikasi web yang berisi poin-poin yaitu pembuatan aplikasi web dengan pendekatan aplikasi desktop, mengedepankan kecepatan akses, kemudahan

pemakaian untuk pengguna, dan interaksi secara *real-time*. Konsep ini menggabungkan fungsionalitas antarmuka dari aplikasi desktop pada aplikasi web dan memanfaatkan teknologi komunikasi untuk keinteraktifannya. Sehingga aplikasi dengan konsep RIA memberikan pengalaman responsif, efektif, dan intuitif untuk pengguna [2].

Dengan mengimplementasikan konsep RIA pada implementasi klien SIP berbasis web akan memudahkan pengguna dalam melakukan komunikasi melalui VoIP serta turut mempopulerkan VoIP sebagai alternatif metode komunikasi yang murah. Di sisi lain, terdapat kekurangan yang dimiliki oleh konsep RIA yaitu ketergantungan pada penggunaan aplikasi pengaya peramban dari pihak ketiga seperti *Flash* dan *Silverlight*.

Standar HTML5 dibuat untuk mengatasi hal tersebut. Standar HTML5 yang merupakan pengembangan lanjut dari standar HTML, menyempurnakan elemen-elemen lama dan menambah API baru untuk pembuatan aplikasi web yang lebih kompleks [3]. Penggunaan standar HTML5 pada implementasi klien SIP berbasis web ini diharapkan dapat menghilangkan penggunaan aplikasi tambahan yang selama ini digunakan untuk membuat RIA.

II. KAJIAN PUSTAKA

A. Voice over Internet Protocol (VoIP)

Voice over Internet Protocol (VoIP) adalah sebuah sistem yang dapat melakukan transmisi suara sebagai paket data melalui *internet protocol* (IP). VoIP yang menggunakan *packet-switching protocol* yang dapat berjalan di berbagai jenis jaringan, seperti jaringan *public*, *private*, kabel, bahkan *wireless* [4].

VoIP terdiri dari beberapa unsur yang membentuk sistem tersebut yaitu:

- 1) *User agent* merupakan perangkat lunak maupun perangkat keras yang bertugas melakukan dan menerima panggilan.
- 2) *Proxy* yang menghubungkan antar pengguna dan melakukan *routing* panggilan.
- 3) Protokol merupakan serangkaian aturan yang wajib dipenuhi oleh *user agent* maupun *proxy* agar komunikasi dapat terjadi. Beberapa protokol pensinyalan antara lain SIP, H.323 dan IAX2.

- 4) Codec merupakan kompresi data audio dan video pada komunikasi VoIP. Beberapa contoh codec antara lain G.711, G.726 dan iSAC untuk audio serta H.264 dan VP8 untuk video.

B. HTML5

HTML5 (*Hypertext Markup Language version 5*) adalah sebuah bahasa markah yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada internet. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan elemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi web yang lebih kompleks [3].

Fitur-fitur yang digunakan pada implementasi klien SIP ini antara lain *HTML5 Video*, *Websocket* dan *WebRTC*. HTML5 Video adalah elemen baru pada HTML5 yang bertujuan untuk menampilkan dan memainkan berkas video. *Websocket* adalah implementasi koneksi socket yang berjalan diatas protokol HTTP yang menyediakan aliran data secara real-time antara klien dan server [5]. *WebRTC* (*Web Real-Time Communication*) adalah sebuah antarmuka pemrograman perangkat lunak yang menelusuri kemungkinan komunikasi data audio dan video antar pengguna melalui perambah [6].

C. Pustaka JavaScript Sip.js

Sip.js adalah sebuah pustaka *JavaScript* yang menyediakan implementasi protokol SIP untuk digunakan dalam pembuatan *user agent* VoIP berbasis web. Pustaka ini dapat membantu dalam pembuatan *user agent* yang dapat melakukan komunikasi data video dan audio pada aplikasi perambah [7].

D. Node.js

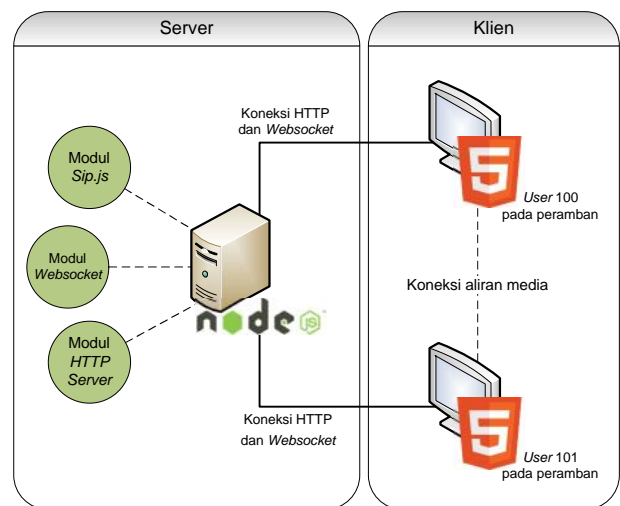
Node.js adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Aplikasi ini ditulis dalam bahasa *JavaScript*, menggunakan basis *event* dan *asynchronous I/O*. Tidak seperti kebanyakan bahasa *JavaScript* yang dijalankan pada perambah, Node.js dieksekusi sebagai aplikasi server. Aplikasi ini terdiri dari *V8 JavaScript Engine* buatan Google dan beberapa modul bawaan yang terintegrasi [8].

Modul-modul yang digunakan dalam implementasi klien SIP ini antara lain Sip.js sebagai implementasi protokol SIP pada Node.js, *Websocket-Node* yang merupakan implementasi *Websocket* pada Node.js dan *Express* yang merupakan kerangka kerja HTTP pada Node.js.

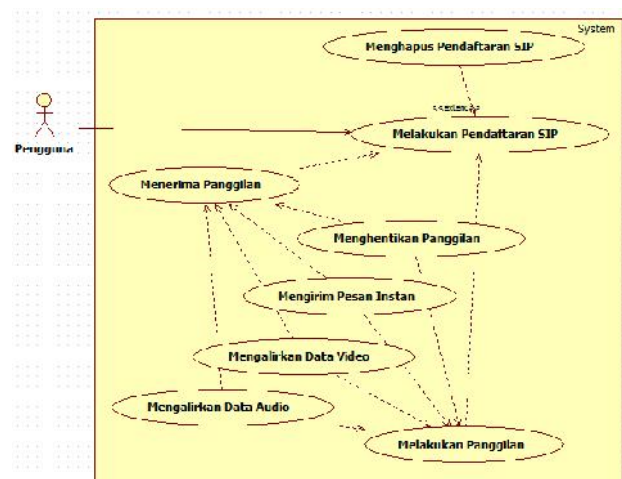
III. DESAIN SISTEM

A. Deskripsi Umum

Pada Tugas Akhir ini akan dikembangkan suatu sistem klien-server SIP berbasis web. Aplikasi klien dikembangkan memakai bahasa markah HTML5. Sedangkan pada sisi server, digunakan *Node.js* yang berbasis bahasa *JavaScript*. Pada aplikasi web ini, pengguna dapat melakukan pendaftaran SIP kemudian melakukan panggilan dan mengirim pesan instan.



Gambar 1. Arsitektur perangkat lunak



Gambar 2. Diagram kasus penggunaan

B. Arsitektur Sistem

Rancangan arsitektur dari sistem yang dibuat dapat diilustrasikan dalam diagram Gambar 1. Pada gambar tersebut terdapat dua komponen utama yaitu klien dan server. Pada klien berupa aplikasi perambah yang mendukung HTML5, sementara pada server terdapat aplikasi *Node.js* yang terdiri dari beberapa modul seperti *HTTP Server*, *Sip.js* dan *Websocket-Node*. Server *Node.js* ini melayani semua permintaan pengguna dengan dibantu modul-modul yang ditunjuk. Modul *HTTP Server* menangani pengambilan laman HTML, modul *Websocket-Node* menangani koneksi *Websocket* sementara modul *Sip.js* menangani pesan SIP dari pengguna. Pesan SIP akan dikirimkan melalui server, sedangkan koneksi aliran media terjadi secara langsung antara pengguna.

C. Deskripsi Singkat Aplikasi

Aplikasi Klien SIP berbasis web ini memiliki fitur-fitur yang diilustrasikan pada diagram kasus penggunaan pada Gambar 2 berikut ini.

- 1) Melakukan Pendaftaran SIP
Pengguna melakukan pendaftaran SIP agar dapat dihubungi dan menghubungi pengguna lain.
- 2) Menghapus Pendaftaran SIP
Data pendaftaran SIP pengguna akan dihapus dari server.
- 3) Melakukan Panggilan
Pengguna menginisiasi sesi panggilan baru dengan pengguna lain.
- 4) Menerima Panggilan
Pengguna menerima undangan sesi panggilan dari pengguna lain.
- 5) Menghentikan Panggilan
Pengguna menghentikan sesi panggilan.
- 6) Mengalirkan Data Video
Pengguna dapat memilih untuk mengalirkan atau tidak mengalirkan data video.
- 7) Mengalirkan Data Audio
Pengguna dapat memilih untuk mengalirkan atau tidak mengalirkan data audio.
- 8) Mengirim Pesan Instan
Pengguna mengirim pesan instan pada pengguna lain.

IV. IMPLEMENTASI

A. Implementasi Transport Websocket pada Modul Sip.js

Modul Sip.js hanya menyediakan transport TCP dan UDP, untuk itu dibutuhkan implementasi modul *Websocket-Node* pada modul Sip.js agar modul ini dapat menerima dan melakukan koneksi *Websocket*.

B. Implementasi Proses Pendaftaran SIP

Proses pendaftaran SIP terdiri dari proses melakukan pendaftaran dan proses menghapus pendaftaran. Proses ini penting karena pengguna yang terdaftar pada server akan dapat melakukan dan menerima panggilan. Proses-proses ini diimplementasikan pada klien dan server.

C. Implementasi Proses Panggilan

Proses panggilan terdiri dari proses melakukan panggilan, menerima panggilan dan menutup panggilan. Proses-proses tersebut merupakan proses-proses utama dalam aplikasi web ini. Proses-proses ini diimplementasikan pada klien dan server.

D. Implementasi Proses Mengirim Pesan Instan

Proses ini memanfaatkan pesan permintaan MESSAGE pada SIP untuk memungkinkan pengguna mengirim pesan instan pada pengguna lain. Proses ini diimplementasikan pada klien dan server.

E. Implementasi Proses Mendeteksi Kompatibilitas Peramban

Proses ini adalah proses yang dilakukan oleh klien untuk mengecek kemampuan peramban dalam menjalankan fitur-fitur HTML5 yaitu *HTML5 Video*, *Websocket* dan *WebRTC*.



Gambar 3. Uji coba proses menerima panggilan

V. UJI COBA DAN EVALUASI

A. Uji Coba Fungsionalitas Sistem

Pada uji coba fungsionalitas ini, fitur aplikasi klien SIP berbasis web sudah dapat berjalan semua sesuai dengan diagram kasus penggunaan pada Gambar 2 yang sudah dirancang pada bab sebelumnya. Gambar 3 menunjukkan uji coba menerima panggilan yang sukses, di mana sesi komunikasi berhasil dibuat.

B. Uji Coba Kompatibilitas Peramban

Uji coba ini menguji kemampuan beberapa peramban populer dalam menggunakan fitur-fitur HTML5 yang dalam hal ini adalah *HTML5 Video*, *Websocket* dan *WebRTC*. Aplikasi peramban yang diuji yaitu Google Chrome Developer Release v21.0.1163.0 dev-m, Mozilla Firefox v13.0, Safari v5.17, Opera v12.0.1467 dan Internet Explorer v9.0.8112. Berikut pada Tabel 1 adalah hasil dari uji coba tersebut.

Tabel 1.
Hasil uji coba kompatibilitas peramban

Aplikasi Peramban	HTML5 Video	Websocket	WebRTC
Google Chrome Developer Release	0	0	0
Mozilla Firefox	0	0	x
Safari	x	0	x
Opera	0	x	x
Internet Explorer	0	x	x

C. Uji Coba Quality of Service

Pada uji coba ini dilakukan pengukuran *Quality of Service* yang terdiri dari pengukuran *delay*, *jitter*, *loss packet* dan *bandwidth* yang dipakai *throughput*. Pengujian ini dilakukan dengan menggunakan kakas *Wireshark* yang dapat menangkap aliran data yang terjadi pada perangkat komputer. Pada uji coba ini, *Wireshark* diletakkan pada masing-masing komputer pengguna.

Pengukuran *delay* dihitung dari waktu paket ditangkap pada komputer penerima dikurangi dengan waktu paket ditangkap pada komputer pengirim. *Jitter* diukur dari selisih antar *delay*

yang terjadi. *Loss packet* diukur dari jumlah paket dengan sumber komputer pengirim yang ditangkap pada komputer pengirim dikurangi dengan komputer penerima. Sementara *bandwidth* yang dipakai diukur dari ukuran keseluruhan paket yang ditangkap pada satu komputer dalam bita dibagi dengan waktu keseluruhan penangkapan paket dalam detik.

Pengujian akan dilakukan tiga kali yaitu dengan skenario dua komputer klien berada pada *subnet* yang sama dan *subnet* yang berbeda, serta enam komputer klien berada pada *subnet* yang tertutup. Berikut pada Tabel 2, Tabel 3 dan Tabel 4 adalah hasil dari uji coba tersebut.

Tabel 2.

Hasil uji coba QoS dengan dua klien pada *subnet* yang sama

Data Tangkapan Paket	
Rata-Rata <i>Bandwidth</i> yang dipakai/ <i>Throughput</i>	102229,3035 bita/detik
Rata-Rata <i>Delay</i>	0,002594 detik
Rata-Rata <i>Jitter</i>	0,001435detik
Rata-Rata <i>Loss Packet</i>	0,0125%

Tabel 3.

Hasil uji coba QoS dengan dua klien pada *subnet* yang berbeda

Data Tangkapan Paket	
Rata-Rata <i>Bandwidth</i> yang dipakai/ <i>Throughput</i>	64033,9005 bita/detik
Rata-Rata <i>Delay</i>	0,023041 detik
Rata-Rata <i>Jitter</i>	0,010113 detik
Rata-Rata <i>Loss Packet</i>	0,6973%

Tabel 4.

Hasil uji coba QoS dengan enam klien pada *subnet* yang tertutup

Data Tangkapan Paket	
Rata-Rata <i>Bandwidth</i> yang dipakai/ <i>Throughput</i>	237698,684 bita/detik
Rata-Rata <i>Delay</i>	0,018649 detik
Rata-Rata <i>Jitter</i>	0,010552 detik
Rata-Rata <i>Loss Packet</i>	0,9318%

D. Uji Coba Kualitas Audio dan Video

Pada uji coba ini akan dilakukan pengukuran kualitas audio dan video melalui survei yang diberikan pada pengguna. Survei ini berisi skor MOS (*Mean Opinion Score*) serta saran yang diberikan untuk kualitas audio dan video yang diterima pengguna. Perlu diperhatikan bahwa pada aplikasi web ini *audio codec* yang digunakan adalah iSAC (*internet Speech Audio Codec*) dan *video codec* yang digunakan adalah VP8.

Survei berhasil dilakukan terhadap 15 orang pengguna dan mendapatkan rata-rata skor MOS untuk *codec* audio iSAC yang digunakan pada aplikasi web ini yaitu 3,8. Berdasarkan pendapat dan saran pengguna, beberapa pengguna menemui *feedback* dan *noise* pada audio yang dihasilkan oleh aplikasi web. Untuk kualitas video, rata-rata pengguna mengatakan kualitas video sudah cukup bagus, tidak terdapat banyak *delay* pada video yang dihasilkan aplikasi web.

VI. KESIMPULAN

Kesimpulan yang dapat ditarik dari implementasi klien SIP berbasis web ini adalah bahwa HTML5 dan Node.js mampu diimplementasikan sebagai aplikasi web klien SIP dan *video chat*. Komunikasi audio dan video yang menggunakan *codec*

iSAC dan VP8 menggunakan *bandwidth* rata-rata dengan nilai terbesar pada komunikasi enam klien subnet yang tertutup, kemudian komunikasi dua klien subnet yang sama, dan paling kecil pada komunikasi dua klien subnet yang berbeda. Hal ini dipengaruhi banyak hal, antara lain banyaknya pengguna dan banyaknya *hop* yang dilalui paket. Survei pengguna terhadap kualitas audio menghasilkan nilai MOS sebesar 3,8. Sedangkan berdasarkan hasil pengujian kompatibilitas peramban yang menguji kemampuan beberapa peramban populer dalam menggunakan fitur-fitur HTML5 yang dibutuhkan oleh aplikasi web ini hanya aplikasi peramban *Google Chrome Developer Release* yang dapat menjalankan aplikasi web ini secara penuh. Sedikitnya aplikasi peramban yang mendukung standar HTML5 dikarenakan standar tersebut masih dalam tahap pengembangan. Aplikasi web ini perlu dikembangkan lebih lanjut dengan mengikuti arah pengembangan standar HTML5, terutama WebRTC, sehingga pada saat standar HTML5 telah diimplementasikan sepenuhnya diharapkan aplikasi web ini dapat dijalankan pada aplikasi peramban apapun pada perangkat apapun, bahkan pada perangkat *mobile*.

UCAPAN TERIMA KASIH

Penulis M.I. mengucapkan terima kasih kepada Tuhan Yang Maha Esa, kedua orangtua dan keluarga penulis, dosen pembimbing, dosen dan kepala jurusan Teknik Informatika, kerabat dekat, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

DAFTAR PUSTAKA

- [1] Van Meggelen, Jim, Leif Madsen, Jared Smith. *Asterisk: The Future of Telephony, Second Edition*. Sebastopol : O'Reilly Media, (2007).
- [2] David, Matthew. *HTML5: Designing Rich Internet Applications*. Burlington : Focal Press, (2010).
- [3] Mavrody, Sergei. *HTML5 & CSS3: Quick Reference*. Chicago : Belisso, (2012).
- [4] Aksara, Frenda Angga. *Pengembangan VoIP Phone Berbasis Web*. Surabaya : Tugas Akhir : Teknik Informatika Institut Teknologi Sepuluh Nopember, (2011).
- [5] Fette, dkk. RFC6455 The Websocket Protocol. [Daring] Desember 2011. <http://tools.ietf.org/html/rfc6455>. Diakses pada tanggal: 27 Juni 2012.
- [6] Ericsson, dkk. WebRTC 1.0: Real-time Communication Between Browsers. [Daring] 30 Mei 2012. <http://dev.w3.org/2011/webrtc/editor/webrtc.html>. Diakses pada tanggal: 2 Juni 2012.
- [7] Singh, Kundan. sip-js SIP in JavaScript. [Daring] Intensity Cloud Technologies, 29 Mei 2012. <http://code.google.com/p/sip-js/>. Diakses pada tanggal: 14 Juni 2012.
- [8] What is Node.js? [Daring] O'Reilly Radar, 6 Juli 2011. <http://radar.oreilly.com/2011/07/what-is-node.html>. Diakses pada tanggal: 29 Mei 2012.